

**IN THE SPECIFICATION:**

Please replace the paragraph at page 7, line 5 (First Amended Paragraph) with the following:

Referring to FIG. 2, a process flow diagram for an exemplary portion of application code development in accordance with an aspect of the present invention is shown. Source code 22 comprises at least one source code module 26. Source code 22 may be developed on personal computer 42 10 or developed elsewhere and ported to computer system 10. By way of example and not limitation, source code 22 may be written in FORTRAN, Ada, Cobol, Modula-2, Pascal, Java, Visual Basic, C, C+, C++ or ~~among~~ other programming languages that support parameterized types. ~~may be compiled.~~ Each source code module 26 is compiled by compiler 20 to produce an associated object code module 24.

Please replace the paragraph at page 7, line 19 (Second Amended Paragraph) with the following:

In FIG. 3, a flow diagram of template information extraction routine 30 is shown in accordance with an aspect of the present invention. At step 31, an initial source code module 26 is obtained. At step 32, it is determined whether the initially obtained module uses a template. For example, module A may call for a STACK, which is a type of template. An illustrative type of a STACK is a first last in first out ~~(FIFO)~~ (LIFO) order tabulation with a limited number of placeholders.

Please replace the paragraph at page 7, line 25 (Third Amended Paragraph) with the following:

If at step 32 a module 26 uses one or more templates, then at step 33 template information, such as template type, algorithm, data structure, and the like, is extracted and a corresponding template information file 28

is created with this extracted information. To facilitate identification of template information files 28 for subsequent processing, a filename and predetermined extension is used. By way of example and not limitation, some sample filenames and their extension for template information files 28 are: moduleA.inst, moduleB.inst, and moduleC.inst. If at step 32 this module 26 does not make use of any template, then at step 34 a determination is made whether there are any more modules 26 to process.

Please replace the paragraph at page 9, line 10 (Fourth Amended Paragraph) with the following:

After receipt of template information files 28, template instantiation routine 40 is used to generate object code for template repository 52 as will now be described. At step 41, list 29 of all template information files 28 is generated. At step 42, template information files 28 contained in list 29 are accessed, and template information contained in those template information files 28 is used to ~~provide~~ generate template source code files 56. Notably, it is possible that no new template source code files 56 need be generated after an initial pass at step 42, because there are no templates or because none of the original templates associated with list 29 depend upon other templates. Thus, template source code in source code files 56 is not amended, and corresponding template object code files 54 are generated by compiler 20 from template source code files 56. However, assuming one or more template source code files 56A not initially present in source code files 56 are to be generated, cross compiler 20 may be called to perform generation of such template source code files 56A.